

1. (original) A method for implementing an electronic program guide, the method comprising:

partitioning a data storage area into a plurality of discrete storage areas;

receiving programming information from a source;
and

storing the received programming information, in its entirety, in the discrete storage areas, each discrete storage area storing programming information that is related in accordance with a predefined criterion.

2. (original) The method of claim 1, wherein the programming information comprises information about individual programs.

3. (original) The method of claim 1, wherein the programming information further comprises tokens used to describe the individual programs and a meaning associated with the tokens.

4. (original) The method of claim 2, wherein the predefined criterion comprises a temporal relationship between the individual programs in the received program information.

5. (original) The method of claim 3, wherein the predefined criterion comprises a numeric relationship between token numbers associated with the tokens.

6. (original) The method of claim 1, wherein a size of each data storage area is selected to store program

information about programs to be broadcast over a defined time interval.

7. (original) The method of claim 1, further comprising referencing the information stored in each discrete storage area using a storage area identifier to identify the information within a storage area and an index of storage area identifiers.

8. (original) The method of claim 7, wherein the storage area identifiers form a pointer chain.

9. (original) The method of claim 7, wherein each discrete storage area which stores programming information no longer required is referenced by an empty identifier indicating that said discrete storage area is available for storing new information.

10. (original) The method of claim 9, further comprising periodically determining if the programming information stored in each discrete storage area is relevant; and marking those storage areas containing programming information that is no longer relevant with the empty identifier.

11. (original) The method of claim 10, wherein determining if the programming information stored in each discrete storage area is relevant comprises checking if the programming information is current or not.

12. (original) The method of claim 1, further comprising determining that specific programming information

is required; and requesting said specific programming information from the source.

13. (original) The method of claim 12, wherein determining that specific programming information is required comprises checking if a user has input a request for specific programming information.

14. (original) The method of claim 12, wherein determining that specific programming information is required comprises checking whether the programming information stored in the discrete storage areas is incomplete for want of specific programming information.

15. (original) A system for implementing an electronic program guide, the system comprising:

- input circuitry configured to receive programming information;

- a processor coupled to the input circuitry;

- a program memory coupled to the processor to store a control program which controls operation of the processor; and

- a data storage area coupled to the processor, the data storage area being partitioned into a plurality of discrete storage areas, wherein under control of the control program the processor operates to store programming information received by the input circuitry in the discrete storage areas, each discrete storage area storing programming information that is related in accordance with a predefined criterion.

16. (original) The system of claim 15, wherein the programming information comprises information about individual programs.

17. (original) The system of claim 16, wherein the programming information further comprises tokens used to described the individual programs and meaning associated with the tokens.

18. (original) The system of claim 16, wherein the predefined criterion comprises a temporal relationship between the individual programs in the received programming information.

19. (original) The system of claim 17, wherein the predefined criterion comprises a numeric relationship between token numbers associated with the tokens.

20. (original) The system of claim 15, wherein a size of each data storage area is selected to store the program information about programs to be broadcast over a defined time interval.

21. (original) The system of claim 15, wherein the processor further operates to reference the information stored in each discrete storage area using a storage area identifier to identify the information within a storage area and index of storage area identifiers.

22. (original) The system of claim 21, wherein the storage area identifiers form a pointer chain.

23. (original) The system of claim 21, wherein the processor operates to reference each discrete storage area

which stores programming information no longer required with an empty identifier indicating that said discrete storage area is available for storing new information.

24. (original) The system of claim 23, wherein the processor further operates to periodically determine if the programming information stored in each discrete storage area is relevant; and to mark those storage areas containing programming information that is no longer relevant with the empty identifier.

25. (original) The system of claim 24, wherein the processor operates to determine if the programming information stored in each discrete storage area is relevant by checking if the program information is current or not.

26. (original) The system of claim 15, wherein the processor operates to determine that specific programming information is required; and to request said specific programming information from a source of programming information.

27. (original) The system of claim 26, wherein in determining that specific programming information is required the processor checks if a user has input a request for specific programming information.

28. (original) The system of claim 26, wherein in determining the specific programming information is required, the processor checks the programming information stored in the discrete storage areas is incomplete for want of specific programming information.

29. (original) A system for implementing an electronic guide, the system comprising a processor coupled to a memory having stored thereon a sequence of instructions which when executed by the processor cause the processor to perform a method comprising:

partitioning a data storage area into a plurality of discrete storage area;

receiving programming information from a source;
and

storing the received programming information, in its entirety, in the discrete storage areas, each discrete storage area storing programming information that is related in accordance with a predefined criterion.

30. (original) The system of claim 29, wherein the programming information comprise information about individual programs.

31. (original) The system of claim 30, wherein the programming information further comprises tokens used to describe the individual program and a meaning associated with the tokens.

32. (original) The system of claim 30, wherein the predefined criterion comprises a temporal relationship between the individual programs in the received program information.

33. (original) The system of claim 30, wherein the predefined criterion comprises a numeric relationship between the token numbers associated with the tokens.

34. (original) The system of claim 29, wherein a size of each data storage area is selected to store program information about programs to be broadcast over a defined time interval.

35. (original) The system of claim 29, wherein the method further comprises referencing the information stored in each discrete storage area using a storage area identifier to identify the information within a storage area and index of storage area identifiers.

36. (original) The system of claim 35, wherein the storage area identifiers form a pointer chain.

37. (original) The system of claim 35, wherein each discrete storage area which stores programming information no longer required is referenced by an empty identifier indicating that said discrete storage area is available for storing new information.

38. (original) The system of claim 37, wherein the method further comprises periodically determining if the programming information stored in each discrete storage area is relevant; and marking those storage areas containing programming information that is no longer relevant with the empty identifier.

39. (original) The system of claim 38, determining if the programming information stored in each discrete storage identifier is relevant comprises checking if the programming information is current or not.

40. (original) The system of claim 29, wherein the method further comprises determining that specific programming information is required and requesting said specific programming information from the source.

41. (original) The system of claim 40, wherein determining the specific programming information is required comprises checking if a user has input a request for specific programming information.

42. (original) The system of claim 40, wherein determining that specific programming information is required comprises checking whether the programming information stored in the discrete storage area is incomplete for want of specific programming information.

43. (original) A system for implementing an electronic program guide, the system comprising:

means for partitioning a data storage area into a plurality of discrete storage areas;

means for receiving programming information from a source; and

means for storing the received programming information, in its entirety, in the discrete storage areas, each discrete storage area storing programming information that is related in accordance with a predefined criterion.

44. (original) The system of claim 43, wherein the programming information comprises information about individual programs.

45. (original) The system of claim 44, wherein the program information further comprises tokens used to describe the individual programs and a meaning associated with the tokens.

46. (original) The system of claim 44, wherein the predefined criterion comprises a temporal relationship between the individual programs in the received program information.

47. (original) The system of claim 45, wherein the predefined criterion comprises a numeric relationship between token numbers associated with a token.

48. (original) The system of claim 43, wherein a size of each data storage area is selected to store program information about programs to be broadcast over a defined time interval.

49. (original) The system of claim 43, further comprising means for referencing the information stored in each discrete storage area using a storage area identifier to identify the information within a storage area and index of storage area identifiers.

50. (original) The system of claim 49, wherein the storage area identifiers form a pointer chain.

51. (original) The system of claim 49, wherein each discrete storage area which stores programming information no longer required is referenced by an empty identifier indicating that said discrete storage area is available for storing new information.

52. (original) The system of claim 51, further comprising means for periodically determining if the programming information stored in each discrete storage area is relevant; and marking those storage areas containing program information that is no longer relevant with the empty identifier.

53. (original) The system of claim 52, wherein determining if the programming information stored in each discrete storage area is relevant comprises checking if the programming information is current or not.

54. (original) The system of claim 43, further comprising means for determining that specific programming information is required; and means to request said specific programming information from the source.

55. (original) The system of claim 54, wherein the means for determining the specific programming information is required comprises means for checking if a user has input a request for specific programming information.

56. (original) The system of claim 55, wherein the means to determine the specific programming information is required comprising means for checking whether the programming information stored in the discrete storage is incomplete for want of specific programming information.